# Mobile Malware Security Challenges and Cloud-Based Detection

Nicholas Penning, Michael Hoffman, Jason Nikolai, Yong Wang
College of Business and Information Systems
Dakota State University
Madison, SD 57042
{nfpenning, mjhoffman13054, janikolai}@pluto.dsu.edu, yong.wang@dsu.edu

*Abstract*— **Mobile malware has gained significant ground since the dawning of smartphones and handheld devices. TrendLabs estimated that there were 718,000 malicious and high risk Android apps in the second quarter of 2013. Mobile malware malicious infections arise through various techniques such as installing repackaged legitimate apps with malware, updating current apps that piggy back malicious variants, or even a drive-by download. The infections themselves will perform at least one or multiple of the following techniques, privilege escalation, remote control, financial charge, and information collection, etc. This paper summarizes mobile malware threats and attacks, cybercriminal motivations behind malware, existing prevention methods and their limitations, and challenges encountered when preventing malware on mobile devices. The paper further proposes a cloud-based framework for mobile malware detection. The proposed framework requires a collaboration among mobile subscribers, app stores, and IT security professionals. The cloud-based malware detection is a promising approach towards mobile security.**

***Keywords- mobile, malware, security, detection, cloud, Android***

## I. Introduction

Mobile devices, such as smartphones and tablets, have been widely used for personal and business purposes. According to a recent report from KPBC, the number of smartphone users has risen above a billion in Q3 2012 globally [1]. Gartner estimated that 1.2 billion smartphones and tablets could be sold in 2013 [2].

One of the greatest threats to data privacy and security is mobile malware. As the largest installed base of mobile platform, Android accounted for 81% of all smartphone shipments in Q3 2013 [3]. TrendLabs estimated that there were 718,000 malicious and high risk Android apps in the second quarter of 2013 [4]. In addition, according to F-Secure, out of the 259 new threat families and new variants of existing families discovered in Q3 2013, 252 were Android threats [5]. Statistically, Android is the most targeted mobile platform when it comes to malicious apps. This paper focuses on malware security challenges in Android devices. However, many security issues discussed and the approaches presented in this paper also apply to other mobile platforms.

Mobile malware malicious infections arise through various techniques such as installing repackaged legitimate apps with malware, updating current apps that piggy back malicious variants, or even a drive-by download. The infections themselves will perform at least one or multiple of the following techniques, privilege escalation, remote control, financial charge, and information collection, etc. The previous stated techniques provide a malicious attacker with a variety of options to utilize a compromised mobile device.

Many mobile malware prevention techniques are ported from desktop or laptop computers. However, due to the uniqueness of smartphones [6], such as multiple-entrance open system, platform-oriented, central data management, vulnerability to theft and lost, etc., challenges are also encountered when porting existing anti-malware techniques to mobile devices. These challenges include, inefficient security solutions, limitations of signature-based mobile malware detection, lax control of third party app stores, and uneducated or careless users, etc.

This paper reviews and summarizes mobile malware threats and attacks, cybercriminal motivations behind malware, existing prevention methods and their limitations, and challenges encountered when preventing malware on mobile devices. Collaborate is an effective technique towards future mobile malware detection [7], [8]. The paper further proposes a cloud-based framework for mobile malware detection. The proposed framework requires a collaboration among mobile subscribers, app stores, and IT security professionals. The cloud-based malware detection is a promising approach towards mobile security.

The remainder of the paper is organized as follows: Section II summarizes mobile malware threats and attacks. Section III reveals the cybercriminal motivations behind mobile malware. Section IV reviews and compares existing mobile malware prevention techniques, followed by the discussion of challenges to prevent malware on mobile devices in Section V. Section VI presents the proposed cloud-based mobile malware detection framework. Section VII concludes the paper.

## II. Mobile Malware Threats and Attacks

Mobile phone virus emerged as early as 2004. Since then, significant amounts of malware have been reported in smartphones.

### A. Mobile Malware

Smartphone malware falls in three main categories, virus, Trojan, and spyware [6]. Trojan and spyware are the dominant malware in smartphones.

#### 1) Virus

Virus emerged in mobile phones as early as 2004. They are typically disguised as a game, a security patch, or other desirable applications and are then downloaded to a smartphone.

Viruses can spread not only through internet downloads or memory cards, but they can also spread through Bluetooth. Two Bluetooth viruses have been reported in smartphones: Bluejacking and Bluesnarfing. Bluejacking sends unsolicited messages over Bluetooth to Bluetooth-enabled device (limited range, usually around 33 feet). Bluesnarfing can access unauthorized information in a smartphone through a Bluetooth connection.

#### 2) *Trojan*

Trojan is another type of malware in smartphones. Most Trojans in smartphones are related to activities such as recording calls, instant messages, locating via GPS, forwarding call logs and other vital data. SMS Trojans are one of the largest categories of mobile malware. It runs in the background of an application and sends SMS messages to a premium rate account owned by an attacker. Malware belonging to this category is the HippoSMS. It increases the phone billing charges of users by sending SMS to premium mobiles and also blocks messages from service providers to users alerting them of additional charges.

#### 3) *Spyware*

Spyware collects information about users without their knowledge. Spyware has given rise to many concerns about invasion of users' privacy. According to Juniper's 2011 malware report, spyware was the dominate one of malware which affects Android phones [9]. It accounted for 63 percent of the samples identified in 2011. A concern of Carrier IQ was recently raised. A Carrier IQ application is usually pre-installed in a smartphone device and it collects usage data to help carriers to make network and service improvements. Mobile operators, device manufacturers, and application vendors may need this usage information to deliver high quality products and services. However, smartphone subscribers have to be assured what data is being collected and how said data is processed and stored. Mobile subscribers' privacy needs to be protected when data is transmitted, processed, and stored.

### B. *Threats and Attacks*

Smartphones are under numerous threats and attacks. These threats and attacks are summarized below.

#### 1) *Sniffing*

There are various ways to sniff or tap a smartphone. In 2010, Karsten showed that GSM's encryption function for call and SMS privacy, A5/1, could be broken in seconds [11]. All GSM subscribers are at the risk of sniffing attacks. Further, as eavesdropping software continues to become available and installed in smartphones, smartphone subscribers with 3G or 4G networks are at risk too.

#### 2) *Spam*

Spam can be carried through emails or MMS messages. Spam messages may include URLs which direct users to phishing or pharming websites. MMS spam can also be used for starting denial of service attacks. The number of U.S. spam text messages rose 45 percent in 2011 to 4.5 billion messages, according to Richi Jennings, an industry analyst.

#### 3) *Spoofing*

An attacker may spoof the "Caller ID" and pretend to be a trusted party. Researchers also demonstrated how to spoof MMS messages that appeared to be messages coming from 611, the number the carriers use to send out alerts or update notifications [10]. Further, base stations could be spoofed too [11].

#### 4) *Phishing*

Phishing attack is a way to steal personal information, such as user name, password, credit card account, etc., by masquerading as a trusted party. Many phishing attacks have been recognized in social networking, emails, and MMS messages. For example, many mobile applications include social sharing and payment buttons. A malicious application can similarly include a "Share on Facebook" button and redirect the users to a spoofed target application. The target application can then request the user's secret credentials and steal the data.

#### 5) *Pharming*

In pharming attacks, attackers can redirect web traffic in a smartphone to a malicious or bogus website. By collecting the subscriber's smartphone information, specific attacks may follow after pharming attacks. For example, when a user browses a web site in a smartphone, the HTTP header usually includes the smartphone's operating system, browser information, and version number. With this information, an attacker may learn the security leaks of the smartphone and is then able to start specific attacks on the smartphone.

#### 6) *Vishing*

Vishing is a short term for "voice" and "phishing". It is an attack which malicious users try to gain access to private and financial information from a smartphone subscriber. By spoofing the "Caller ID", the attacker may look like from a trusted party and spoof the smartphone users to release their personal credentials.

#### 7) *Data leakage*

Data leakage is the unauthorized transmission of personal information or corporate data. It includes both intentional and unintentional data leakage. Malicious software may steal person's information such as contact list, location information, and bank information and send this data to a remote website. A smartphone owner may be at risk of identity theft due to the data leakage from the phone. Business owners or classified users such as government and military users have even more concerns about data leakage. ZitMo, a mobile version of Zeus, has been found in Symbian, BlackBerry and Android and could be used to steal one-time passwords sent by banks to authenticate mobile transactions.

### *8) Vulnerabilities of Webkit engine*

A vulnerability on web browsers in smartphones is another usual scenario of attacks. The Webkit engine used by almost all mobile platforms may include vulnerabilities which allow attackers to crash user applications and execute malicious code. In a recent vulnerability revealed by CrowdStrike, the attackers could use the Webkit vulnerability to install a remote access tool to eavesdrop on smartphone conversations and monitor the user locations. The vulnerability has been found in BlackBerry, iOS and Android.

### *9) Denial of Service (DoS) attacks*

Smartphone users also suffer from various DoS attacks.

- *Jamming attacks* Smartphones are based on radio communication technology and they are vulnerable to jamming attacks. The communication between smartphones and base stations could be disrupted using jamming devices.
- *Flooding attacks* Flooding attacks can be carried out using both text messages or incoming calls. A smartphone could be disabled if it received hundreds of text messages or incoming calls.
- *Exhaustion attacks* Battery exhaustion attack is another DoS attack on a smartphone which causes more battery discharge than is typically necessary.
- *Blocking attacks* Blocking features in a smartphone can be used to start DoS attacks too. If a malicious user keeps calling a smartphone user using a blocked phone number, the smartphone subscriber cannot do anything else.

Many attacks could be turned on in a stealth mode. Users may not observe and realize these attacks for days and months. A malicious user can always plant malware in a smartphone first and use it when in need.

## III. Mobile Malware Cybercriminal Motivations

The cybercriminal motivations behind mobile malware may vary from collecting confidential data to financial gain. The three main motivations behind mobile malware include obtaining financial gain, collecting sensitive data, and accessing private networks.

### *A. Obtain Financial Gain*

The most well-known goal for malware authors is to obtain financial gain. A malicious app has a variety of different profitable possibilities to obtain financial gain. Compromising a mobile device to send out SMS messages to premium rate phone numbers is one of these forms. Generally, a user would text a specific message to a given number and receive some type of service as simple as a ringtone. When the user sends out this message, the message will be forwarded by the service provider to an aggregator or middleman who will send the message back to the user asking if the user wants to approve the purchase. Once the purchase is approved, the user receives the ringtone and is then billed. A malicious approach of this same scenario would be an infected application sends out the message to the aggregator. When the confirmation is received on the user's mobile device, the malicious app accepts the confirmation without asking the user for permission. The message that was sent out usually goes to a malicious number that creates profit for the attacker. Once the user is billed, the malicious attacker gains the amount of money in which they have specifically set for the premium rate number. Attackers often get away with their malicious activities because the end users that are getting billed do not notice the minor charges.

Another way to obtain financial gain is through contact lists. Contact lists often house loads of information, such as, email addresses, phone numbers, birthdates, etc., which is ideal for spammers. An attacker could use a malicious app to collect contact lists on mobile devices and then sell them to spammers in underground markets.

In addition, financial gain for a malicious attacker can also be done through ad revenue. The attack may host a website that has ads which are generating revenue per visit. The attacker could embed links to this ad revenue generated website inside of a mobile app and create multiple requests from any users who have installed the malicious app.

### *B. Collect Sensitive Data*

Smartphones and handheld devices are data-centric devices. The potential data that an attacker may access on a mobile device is incredible. Data housed on mobile devices that is highly targeted includes contact lists, keyboard cache (autocorrect, dictionaries, passwords, etc.), personally identifiable information (SSN, bank account, etc.), locations visited, and user account credentials (email addresses, usernames, and passwords). This data is stored on mobile devices waiting to be harvested. A simple key logger could be installed on a mobile device to capture inputs from a user. The data collected by an adversary could be used for identity theft, sold in an underground market, or used to torment a specific user.

### *C. Access Private Networks*

As Bring Your Own Devices (BYODs) become popular in enterprise environment, an attacker could also use mobile malware to exploit and access a victim's private network [12]. Once the victim's network is compromised, the attacker could access corporate resources, steal corporate data, or use the resources of the network to join a botnet to perform denial of service attacks. An attacker could also use the victim's network to perform other malicious activities to cover their own tracks. This method will utilize the victim's wireless carrier to carry out attacks for the malicious user in a way that the attacks would be tied back to the infected victim's network and not the attacker.

## IV. MOBILE MALWARE PREVENTION METHODS

Zhou *et al.* at North Carolina State University took 1260 collected malware samples and looked at what the top twenty permissions were [13]. They also analyzed the first top free 1260 benign apps and obtained those top twenty permissions. They compared the permissions and found that malicious apps tend to request SMS permissions more frequently, such as READ_SMS, WRITE_SMS, RECEIVE_SMS, and SEND_SMS.

TABLE I. PERMISSIONS REQUESTED BY MALWARE

| | **Num. of Samples** | **%** |
|---|---|---|
| Read_SMS | 790 | 62.70% |
| Write_SMS | 658 | 52.22% |
| Receive_SMS | 499 | 39.60% |
| Write_Contacts | 374 | 29.68% |
| Write_App_Settings | 349 | 27.70% |

The permissions in TABLE I. are found in the top 20 permissions requested by malware but not found in the top 20 permissions requested by benign apps (total samples: 1260). This would leave one to believe that these specific permissions are being used solely for malicious purposes.

A simple solution to detect malware could be based on permissions requested by a mobile app. For example, take every single malware sample known, sort out the independent variants, analyze all of the permissions, and calculate an algorithm that detects what permissions are generally utilized together in a malicious app and compare that to the benign apps. However, this approach may deem ineffective. Mobile apps may constantly change what permissions are being used and sometimes even hiding some permissions from the end user. The remaining of this section reviews a few techniques that may be used to detect and prevent mobile malware.

### A. *Signature-based Detection*

Signature-based malware detection is one of the current malware detection methods. By analyzing known malware results, this approach helps prevent from the known malicious apps to be installed. The issue with signature-based detection is that apps could change through updated code or modified just enough to throw off the signature for the anti-malware application to detect. This approach will catch known malware, but fails to stop new or unknown variants in the wild.

### B. *Google Play Store (Bouncer)*

Google has introduced a new method of detecting malicious apps before they hit the Google Play Store. Bouncer is a new mobile malware detector that Google has been using to scan apps before they hit the app market [14]. Bouncer has the approach to take newly developed applications and determine if they attempt to send SMS out to malicious sites. This technique is great for the apps that are downloaded through the Google Play Store, but is disadvantageous for the users who use third party app stores.

### C. *Manufacture Built-in Security*

With Samsung's new line of Android smartphones, Samsung released a security system known as Samsung KNOX [15]. KNOX addresses platform security with a comprehensive three-pronged strategy to secure the system, i.e., Customizable Secure Boot, ARM TrustZone-based Integrity Measurement Architecture, and a kernel with built- in security enhancements for Android. The Customizable Secure Boot ensures that only verified and authorized software can run on the phones. The TrustZone-based Integrity Measurement Architecture runs in the secure-world and provides continuous integrity monitoring of the Linux kernel. If the software notices that the boot loader has been violated, it takes actions in response such as disabling the kernel and powering down the device. These security enhancements provide a mechanism to enforce the separation of information based on the confidentiality and integrity requirements. In addition to securing the system, KNOX also includes an application known as Samsung KNOX container. This application provides a secure environment within the mobile devices allowing users to protect against data leakage.

### D. *Security Awareness Training*

User's knowledge about malicious activities is arguably one of the strongest prevention methods of downloading and installing malicious apps. Educated mobile subscribers should be able to notice when specific anomalies occur on their smartphones and what needs to be done to mitigate a potential infection. Users with a security mindset will usually backup the data on their devices so that they may mitigate any type of malicious activities by performing a factory reset on their devices to remove any potential dangers. Learning to backup and identify malicious activities and permissions is a current prevention method that needs to be recognized to all of the high risk mobile users.

## V. MOBILE MALWARE PREVENTION CHALLENGES

Many mobile malware prevention techniques are ported from desktop or laptop computers. However, due to the uniqueness features of smartphones [6], such as multiple-entrance open system, platform-oriented, central data management, vulnerability to theft or lost, etc., challenges are also encountered when porting anti-malware techniques to mobile devices. These challenges include, inefficient security solutions, limitations of signature-based mobile malware detection, lax control of third party app stores, and uneducated or careless users.

### A. *Inefficient Security Solutions*

Client side security solutions include anti-virus or anti-malware apps installed on mobile devices to protect against known signatures of malicious apps. However, installing an application to provide real time protection on a mobile device often decreases its performance and battery life. The stereotypical age of the client side user also greatly affects the usefulness of the installed application. For the negligence of keeping the app updated or ignoring specific alerts, this makes client side security solutions a bit ineffective if a user is under twelve years of age or above the age of sixty. Wording also plays a key part in the selection of client based security solutions. A user typically will not search for a specific product but rather something with the words anti-virus or anti-malware.

### B. Limitations of Signature-based Mobile Malware Detection

Another issue for having a large selection of anti-malware programs to choose from is the signagure definitions for which they utilize to find infections. Each security solution will most likely have a different database to look for signatures. Because of that reason, one solution will not protect you from threats that another solution could. Signature-based detection is not efficient enough to protect against even well-known exploits. The reason stems from simple open source programs such as ApkTool, Dex2Jar, and JD (Java Decompiler) that will allow somebody to decompile the packaged APK file, then implement their own code and finally repackage the APK as a different version.

### C. Lax Control of Third Party App Stores

Third party app stores are also available. The Amazon App Store is an app store that was created by Amazon to compete with Google's Play Store. Amazon has a set of guidelines that are used when having an app submitted to the store. When an app is submitted, it goes through the Amazon Mobile App Distribution Portal. This is where Amazon has created a way for developers to submit their apps and follow Amazon's approval process where they go through and test the function of the application.

GETJar is another app store that has applications that range from Android to Apple and any other smartphones. The security process that is required to get an app on the store is similar to what Amazon does. They have the user submit the source code of the application and run it through a number of tests to ensure that it does not breach the terms of services for the store. This store has received great security reviews from multiple security experts.

SlideMe is also a third party application store for Android. The approval process for SlideMe is once again similar to what the other application stores. The submissions are reviewed by SlideMe staff to ensure the applications meet the minimum standards and quality guidelines. These guide lines include the forbidding of malware. This store claims to have more security producers then the Google Play Store has provided.

### D. Uneducated or Careless Users

When it comes to users and the installation of apps and their permissions, some users do not understand what inherent risks come with some permissions. For example, users who download many apps have always seen the permissions screen that list all of the possibilities on what the app could potentially access. Google does a great job of explaining each permission, but the problem lies in the grouping of permissions and how they could maliciously work together. As discussed in the previous section, the SMS related permissions, such as Read_SMS, Write_SMS, and Receive_SMS, are used frequently by malware. An educated user should be able to identify the combination of these three permissions and should then be aware of the potential dangers of the app they wish to install. The age of Android device users could range from five years on up to eighty years of age. The familiarity of permissions could be foreign to many users because they might not be able to read and understand, or they do not fully grasp the downsides of the technologies they are utilizing. On the other hand, many users do not have the time or patience to read through every single permission before they install the app. In today's society, most people want their desired app downloaded, installed and ready to use in the quickest timeframe possible. The carelessness of users by not reading before installing causes many security flaws. This is why a security solution needs to be developed to limit the amount of output to the user, but at the same time allowing the device to be secure.

## VI. Cloud-based Mobile Malware Detection Framework

Being able to detect new threats in the wild quickly and efficiently is the goal of mobile security. However, this is a very challenging issue due to inefficient security solutions, limitations of signature-based mobile malware detection, lax control of third party app stores, and uneducated or careless users. In the remainder of this section, we first review a few promising techniques to prevent mobile malware. Then, we introduce our proposed cloud-based framework to detect and prevent mobile malware.

### A. Futuristic Mobile Malware Security Strategies

#### 1) Anomaly/heuristics based detection

The goal of an anomaly or heuristic based detection approach could include efficiently monitoring apps to detect malicious behavior. For example, if an application starts to invoke a collection of API calls that are known to be malicious, the user could be alerted of a new threat working on their mobile devices. Combining this method with the "Permission Based" detection method could detect an app that has been updated or modified from a remote source that now demonstrates malicious activity. The detection could be done in real time on a mobile device. Further, a mobile app which could efficiently monitor the connections in and out of a mobile device could be able to alert the user when their phones are receiving or sending out data to malcious sites or locations.

#### 2) App ranking system

An app ranking system is another type of detection method that could be utilized. Apps could be determined based on user reviews, researcher reviews, and analyzed reviews. The way the ratings are implemented in Google's Play Store is a great way to see the quality and functionality of the app, as well as some of the issues that some users encounter when using the app. There could potentially be a security tab inside of the Play Store where users and researchers will be able to add their input of how they feel the app ranks securely. However, the issue resides in having multiple app stores. If there is a central place where users can find all of the top ranked apps, users will feel more confident downloading the highly ranked apps.

### 3) *Cloud-based detection*

Cloud-based detection is a concept that is seemingly the future for fast, efficient, and effective mobile security [7]. Having an intelligent system that solely analyzes malware statically and dynamically will prove to be a worthy opponent against the malware authors. The remaining of this section outlines a framework on how a cloud-based security solution could work to detect new threats as well as identifying reoccurring threats.

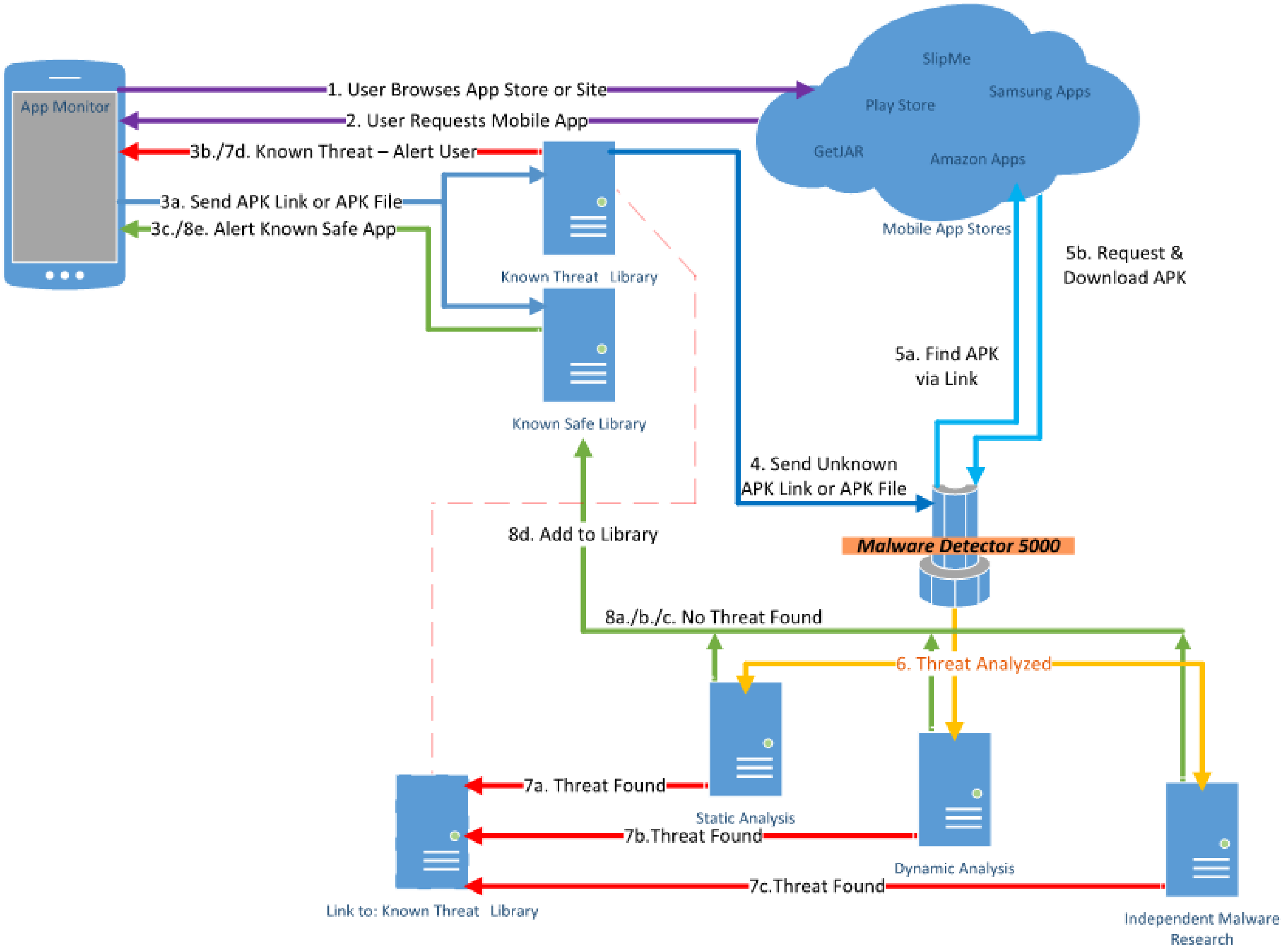
## B. *Cloud-based Detection*

The approach to utilize a cloud service to detect and prevent mobile malware is an attempt to revolutionize the way malicious apps are detected in the wild. A cloud-based mobile malware detection framework is shown in Figure 1. The process starts by requesting an app download from a mobile app store. Then the same request is sent to known threat and known safe libraries which will instantly return a result if the application is found in those libraries. However, if the app is new to both of the libraries, then it will be passed on to the Malware Detector 5000 which will also download the same application. After the download, static and dynamic analysis will be automated to detect any type of threats. The independent malware research virtual machine will house malware that will be explored by human testing. If any threat is found from the static, dynamic or independent research, the app is added to the known threat library which will then be used to alert the user that the app is indeed malicious. If the application is safe throughout the whole process, the app will be added to the safe list and the user will be notified that the app is safe. The details are described below.

### 1) *Components*

The cloud-based detection framework depends on the collaboration of mobile subscribers, app stores, and IT security professionals. The framework consists of:

*App Monitor (a mobile app on the mobile device)*: The purpose of this mobile app is to monitor incoming mobile applications and updates. The requests for new apps and updates are then forwarded to a library of known threat and known safe applications. This application will keep track of apps that have been sent for verification which will act as an



App Monitor
1. User Browses App Store or Site
2. User Requests Mobile App
3b./7d. Known Threat – Alert User
3a. Send APK Link or APK File
3c./8e. Alert Known Safe App
SlipMe
Play Store
Samsung Apps
GetJAR
Amazon Apps
Mobile App Stores
Known Threat Library
Known Safe Library
5b. Request & Download APK
5a. Find APK via Link
4. Send Unknown APK Link or APK File
Malware Detector 5000
8d. Add to Library
8a./b./c. No Threat Found
6. Threat Analyzed
7a. Threat Found
7b.Threat Found
7c.Threat Found
Static Analysis
Dynamic Analysis
Independent Malware Research
Link to: Known Threat Library

Figure 1. Cloud-based mobile malware detection framework

alerting system for known threat and known safe apps. If an app is undergoing malware detection, App Monitor will alert the user that the mobile app is still being analyzed and suggest the user not to install the app.

*Known Threat Library*: The purpose of the known threat library is to house apps that have been flagged as malicious. This will provide quick alerts to users who try and download the malicious apps. This library will contain specific app information such as the date when the malware was detected, what the malware variant is, and the amount of users who have attempted to download and install the app.

*Known Safe Library*: The purpose of the known safe library is to house apps that have been flagged as safe. This will provide quick alerts to users who seek the satisfaction downloading and installing a safe app. This library will also contain specific information on the date when the app was inspected, how many users have downloaded and installed the app, and other relevant information to ensure the safeness of the app.

*Malware Detector 5000*: The purpose of the Malware Detector 5000 is to act as a central station for managing incoming and outgoing apps that are being tested through static analysis, dynamic analysis, and independent research or being stored to the known threat and known safe libraries. Duties include downloading and distributing unknown apps for analysis and transferring discovered threat and safe apps to their appropriate libraries.

*Static Analysis*: Static analysis is the process of analyzing an application without executing the app in an environment. Automated static analysis will review code of an app to find known or suspicious function calls or permissions that deem malicious. With a powerful static analyzer, apps that house known malicious code will be easily spotted and be reported as threats.

*Dynamic Analysis*: Dynamic analysis is the process of analyzing an application while executing the app in a controlled environment. Automated dynamic analysis will monitor network traffic and other communications to catch malicious activity. With a powerful dynamic analyzer, apps that attempt to connect out to unknown or malicious sites, or send SMS messages without authorization will be flagged as malicious and consequently be reported as threats.

*Independent Malware Research*: The purpose of the independent research analysis is allowing human interaction for determining threats in an unknown app. This approach combines static and dynamic analysis and will reveal details that the automated analysis approaches could not. A team of highly experienced malware analysts will work independently to find threats in malicious apps.

### *2) Malware detection procedures*

The steps to detect the malware using the cloud are described as below:

Step 1. A mobile user browses any app store such as Amazon Apps, Google's Play Store, etc.

Step 2. The mobile user then requests an app to download.

Step 3a. The app is sent to the known libraries for malware analysis.

Step 3b. If a known threat is found in the app, the user is alerted that the app is a threat.

Step 3c. If the app is known as safe, the user is notified that the app is safe. If an app is not found in these two libraries, the app is flagged as unknown.

Step 4. Apps that are flagged unknown are transferred to the Malware Detector 5000.

Step 5a. The link to the mobile app is then utilized by finding the APK from the app store or website where it was downloaded.

Step 5b. The Malware Detector 5000 will request and download the unknown app.

Step 6. The Malware Detector 5000 will then supply a sample of the unknown app to a static analysis, dynamic analysis, and independent malware research environments.

Step 7a. If a threat is found via automated static analysis, the app is added to the known threat library, and the user is alerted.

Step 7b. If a threat is found via automated dynamic analysis, the app is added to the known threat library, and the user is alerted.

Step 7c. If a threat is found via independent malware research, the app is added to the known threat library, and the user is alerted.

Step 8a./b./c./d./e. Apps that do not host malicious activities through the whole process will be added to the Known Safe Library, and the user is notified that the application is safe.

### *3) Comparison*

Cloud based detection will allow instant gratification of a known threat or known safe app. If an app is flagged as unknown, the user will have the opportunity to wait a small timeframe to get the app fully analyzed before the app is installed. The originality of the cloud-based framework is the fact that any Android application could be uploaded and reviewed for analysis. If the user decides to install the unknown app anyways, the Malware Detector 5000 will begin the process of investigating the app. Once the app is flagged as safe or a threat, the user will be alerted immediately.

The benefit of having a cloud-based detection approach will place all of the work outside of the mobile device. The mobile device communicates to libraries for assistance on finding out if an app is malicious or safe. This approach will prevent the mobile device from scanning the application on the client side and instead push the scanning onto more powerful and efficient systems. A user will have the opportunity to wait for an app under investigation to be reviewed before trusting just one anti-malware scanner on his/her mobile device.

## VII. Summary

The growth of mobile malware will likely continue to explode as the adoption of mobile devices is still in its early stage. A few mobile malware prevention techniques exist and commercial products to detect and prevent mobile malware are also available. However, the continuing growth of mobile malware indicates that there are no current effective approaches to detect and prevent mobile malware. Mobile devices have many unique features and raise many security challenges to detect and prevent malware on mobile devices, such as inefficient security solutions, limitations of signature-based mobile malware detection, lax control of third party app stores, and uneducated or careless users. This paper proposes a cloud-based framework for mobile malware detection. The framework requires a collaboration among mobile subscribers, app stores, and IT security professionals. The cloud-based mobile malware detection is a promising approach towards mobile security. Our future work includes more study on the framework and how to utilize cloud services and collaborations for mobile malware detection.

## ACKNOWLEDGEMENTS

This work is partially supported by NSF Grant No. CNS-1337529. The authors thank the anonymous reviewers for their valuable comments on this manuscript.

## REFERENCES

[1] M. Meeker and L. Wu, "Internet Trends," 2013.

[2] Gartner Press Release, "Gartner Says 821 Million Smart Devices Will Be Purchased Worldwide in 2012; Sales to Rise to 1.2 Billion in 2013," Barcelona, Spain, 06-Nov-2012.

[3] IDC, "Android Pushes Past 80% Market Share While Windows Phone Shipments Leap 156.0% Year Over Year in the Third Quarter," 12-Nov-2013.

[4] T. Micro, "Trend Labs 2Q 2013 Security Roundup," 2013.

[5] F-Secure, "Mobile Threat Report July-September 2013," 2013.

[6] Y. Wang, K. Streff, and S. Raman, "Smartphone Security Challenges," *Computer (Long. Beach. Calif).*, vol. 45, no. 12, pp. 52–58, Dec. 2012.

[7] M. Chandramohan and H. B. K. Tan, "Detection of Mobile Malware in the Wild," *Computer*, vol. 45, no. 9. pp. 65–71, 2012.

[8] L. Yang, V. Ganapathy, and L. Iftode, "Enhancing Mobile Malware Detection with Social Collaboration," *2011 IEEE Third Int'l Conf. Privacy, Secur. Risk Trust 2011 IEEE Third Int'l Conf. Soc. Comput.*, pp. 572–576, 2011.

[9] Juniper Neworks, "2011 Mobile Threats Report," 2012.

[10] Z. Lackey and L. Miras, "Attacking SMS," *BlackHat 2009*, 2009.

[11] D. Perez and J. Pico, "A practical attack against GPRS/EDGE/UMTS/HSPA mobile data communications," *Black Hat DC*, 2011.

[12] Y. Wang, J. Wei, and K. Vangury, "Bring Your Own Device Security Issues and Challenges," in *The 11th Annual IEEE Consumer Communications & Networking Conference*, 2014.

[13] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution," *Secur. Priv. (SP), 2012 IEEE …*, no. 4, pp. 95–109, 2012.

[14] H. Lockheimer, "Android and Security," *Google Mobile Blog*, 2012.

[15] Samsung, "Samsung KNOX." [Online]. Available: http://www.samsung.com/global/business/mobile/solution/security/samsung-knox. [Accessed: 01-Jan-2014].